Rethinking the Role of Pre-Trained Networks
in Source-Free Domain Adaptation

Wenyu Zhang[1], Li Shen[1], Chuan-Sheng Foo[1,2]

[1]Institute for Infocomm Research (I^{2}R), Agency for Science, Technology and Research (A*STAR)

[2]Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A*STAR)

Abstract

Source-free domain adaptation (SFDA) aims to adapt a source model trained on a fully-labeled source domain to an unlabeled target domain. Large-data pre-trained networks are used to initialize source models during source training, and subsequently discarded. However, source training can cause the model to overfit to source data distribution and lose applicable target domain knowledge. We propose to integrate the pre-trained network into the target adaptation process as it has diversified features important for generalization and provides an alternate view of features and classification decisions different from the source model. We propose to distil useful target domain information through a co-learning strategy to improve target pseudolabel quality for finetuning the source model. Evaluation on 4 benchmark datasets show that our proposed strategy improves adaptation performance and can be successfully integrated with existing SFDA methods. Leveraging modern pre-trained networks that have stronger representation learning ability in the co-learning strategy further boosts performance.

1. Introduction

Deep neural networks have demonstrated remarkable ability in diverse applications, but their performance typically relies on the assumption that training *(source domain)* and test *(target domain)* data distributions are the same. This assumption can be violated in practice when the source data does not fully represent the entire data distribution due to difficulties of real-world data collection. Target samples distributed differently from source samples (due to factors such as background, illumination and style [13, 20]) are subject to *domain shift* (also known as *covariate shift*), and can severely degrade model performance.

Domain adaptation (DA) aims to tackle the domain shift problem by transferring knowledge from a fully-labeled source domain to an unlabeled target domain. The classic setting of *unsupervised domain adaptation* assumes source

and target data are jointly available for training [45]. Motivated by the theoretical bound on target risk derived in [3], a key strategy is to minimize the discrepancy between source and target features to learn domain-invariant features [11, 27, 24, 42, 52, 14, 17, 12, 47]. However, access to source data can be impractical due to data privacy concerns. Recently, the *source-free domain adaptation* (SFDA) setting is proposed [28] to split adaptation into two stages: (a) training network with fully labeled source data, and (b) adapting source model with unlabeled target data. An example use case in a corporate context is when the vendor has collected data to train a source model, and clients seek to address the same task for their own environments, but data sharing for joint training cannot be achieved due to proprietary or privacy reasons. The vendor makes available the source model, which the clients can adapt with available resources. In this work, we assume the role of the clients.

We focus on the classification task where source and target domain share the same label space. The source model is typically obtained by training a selected network on source data with supervised loss. For adaptation, existing SFDA methods generate or estimate source-like representations to align source and target distributions [38, 8], make use of local clustering structures in the target data [49, 48, 50], and learn semantics through self-supervised tasks [46, 29]. [28, 23, 25] use the source model to generate target pseudolabels for finetuning, and [19, 5, 29] further select samples with the low-entropy or low-loss criterion. However, model calibration is known to degrade under distribution shift [35]. We observe in Figure 1 that target pseudolabels produced by source model can be biased, and outputs such as prediction confidence (and consequently entropy and loss) may not reflect accuracy and cannot reliably be used alone to improve pseudolabel quality.

We reconsider the role of pre-trained networks in SFDA. Due to their common usage, the discussion in this paper is within the scope of ImageNet networks, but can be applied to other large data pre-trained networks. As in Figure 2, ImageNet weights are conventionally used to initialize source models and subsequently discarded. However,
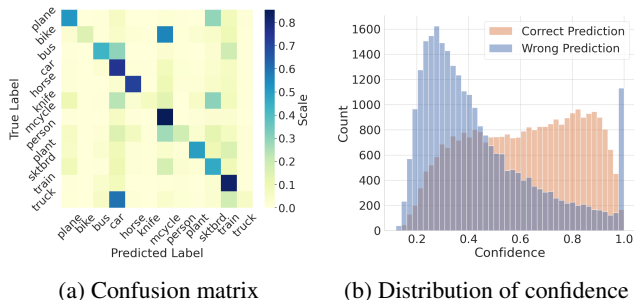
(a) Confusion matrix   (b) Distribution of confidence

Figure 1: VisDA-C source-trained ResNet-101 produces unreliable pseudolabels on target samples, and is overconfident on a significant number of incorrect predictions.

ImageNet networks have diverse features important for generalization [6], and finetuning on source data can overfit to source distribution and potentially lose pre-existing target information. Furthermore, modern ImageNet networks may have better generalizability following new architectures and training schemes. We seek to answer the questions:

- *Can finetuning pre-trained networks on source data lose applicable target domain knowledge?*
- *Given a source model, whether and how pre-trained networks can help its adaptation?*

We propose to integrate ImageNet networks into the target adaptation process, as depicted in Figure 2, to distil any effective target domain knowledge they may hold. They can help to correct source model prediction bias and produce more accurate target pseudolabels to finetune the source model. We design a simple two-branch co-learning strategy where the adaptation and pre-trained model branch iteratively updates to collaboratively generate more accurate pseudolabels, which can be readily applied to existing SFDA methods as well. We provide an overview of our strategy in Figure 3, and summarize our contributions:

- We observe that finetuning pre-trained networks on source data can lose generalizability for target domain;
- Based on the above observation, we propose integrating pre-trained networks into SFDA target adaptation;
- We propose a simple co-learning strategy to distil useful target domain information from a pre-trained feature extractor to improve target pseudolabel quality;
- We demonstrate performance improvements by the proposed strategy (including just reusing the ImageNet network in source model initialization) on 4 benchmark SFDA image classification datasets.

2. Related Works

2.1. Unsupervised domain adaptation

In traditional unsupervised DA, networks are trained jointly on labeled source and unlabeled target dataset to op-
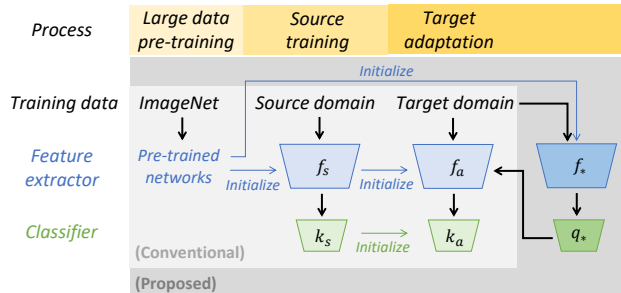

Figure 2: Overview of conventional and proposed frameworks: We propose incorporating pre-trained networks during target adaptation.
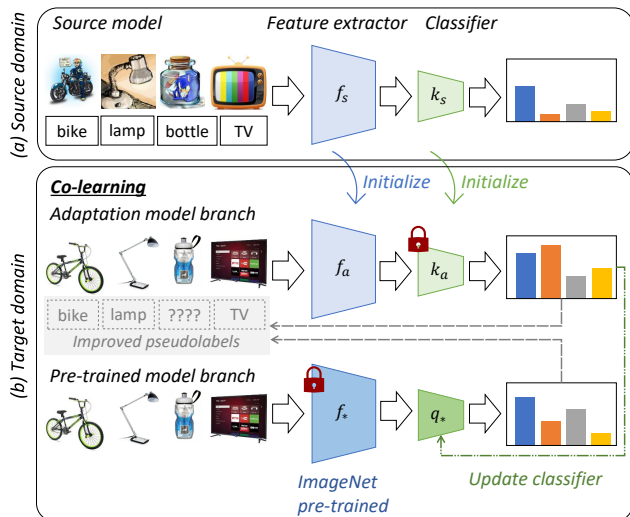

Figure 3: Overview of proposed strategy: (a) Source model trained on source domain is provided. (b) We adapt the source model through a co-learning strategy where the adaptation model and an ImageNet pre-trained model collectively produce more reliable pseudolabels for finetuning. Note since the ImageNet feature extractor is frozen, the strategy can be implemented with a feature bank.

timize task performance on the target domain [45]. A popular strategy is to learn domain-invariant features via minimizing domain discrepancy measured by a inter-domain distance or adversarial loss [11, 27, 24, 42, 52, 14, 17, 12, 47]. [33] learns a distribution of classifiers to better identify local regions of misalignment between source and target domains, and [7, 34] facilitate alignment between distant domains by generating intermediate domains. [26] augments source data to assume target style. Other methods encourage target representations to be more class-discriminative by learning cluster structure [43] or minimizing uncertainty [16]. Amongst confidence-based methods, [12] trains a spherical neural network to select target samples for pseudolabeling, [34, 10] selects high confidence predictions as positive pseudolabels and [10] applies mean

teacher from semi-supervised learning. Due to the assumed access to both source and target data, these methods are unsuitable when source data is not available for joint training.

2.2. Source-free domain adaptation

In SFDA, the source model is adapted with unlabeled target dataset. [21, 22] train on source domain with auxiliary tasks or augmentations and [40] calibrates uncertainties with source data to obtain improved source models, but these strategies cannot be applied on the client-side where source data is not accessible. Some methods make use of local clustering structures in the target dataset to learn class-discriminative features [49, 48, 50], and learn semantics through self-supervised tasks such as rotation prediction [46, 29]. [39] proposes multi-centric clustering, but the cluster number is not straightforward to select for each dataset. [46] learns a new target-compatible classifier, and [25] generates target-style data with a generative adversarial network to improve predictions. Other methods generates source-like representations [38] or estimates source feature distribution to align source and target domains [8]. [9] and [15] leverage multiple source domains. [28, 23, 25] use the source model to generate pseudolabels for the target dataset, and align target samples to the source hypothesis through entropy minimization and information maximization. To improve pseudolabel quality, [19, 5, 29] select target samples with low entropy or loss for pseudolabeling and finetuning the source model. We find that source model outputs may not reliably reflect target pseudolabel accuracy due to domain shift in Figure 1. We instead make use of an ImageNet pre-trained feature extractor to correct the source bias and produce more reliable pseudolabels.

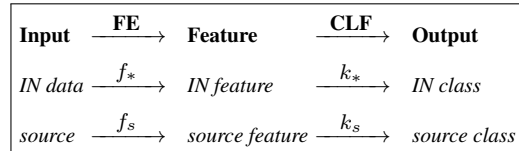## 3. Role of Pre-trained Networks in SFDA

3.1. Conventional role

In the conventional source-free domain adaptation framework in Figure 2, during source training, the source model is initialized with pre-trained weights and then trained on source data with supervised objective. The pre-trained weights are learned from large and diverse datasets such as ImageNet (IN). Transferring from pre-trained networks is a common practice in computer vision to improve the generalization capability of the trained model and to reduce the requirement on the quantity of training data.

3.2. Considerations on target generalizability

In SFDA, the goal is to accurately estimate $p(y_t|x_t)$ for target input $x_t \in \mathcal{X}_t$ and output $y_t \in \mathcal{Y}_t$. For a model with feature extractor (FE) f and classifier (CLF) k and normalizing function σ, the estimated probabilities are $[\hat{p}(y|x_t)]_{y \in \mathcal{Y}_t} = \sigma\left(k(f(x_t))\right)$. We note that the source model may not be the network that maximizes the accuracy

of $\hat{p}(y_t|x_t)$. Large data pre-trained networks, such as the ImageNet (IN) networks used as initializations for source training, may be more compatible with the target domain instead. That is, class-discriminative information useful for the target domain may be lost from the ImageNet network during source training as the source model learns to fit only the source data distribution. We list the mapping from input to feature space, and from feature to output space for the two models below.

$$
\begin{array}{lll}
\textbf{Input} \xrightarrow{\textbf{FE}} & \textbf{Feature} \xrightarrow{\textbf{CLF}} & \textbf{Output} \\[4pt]
IN\ data \xrightarrow{f_*} & IN\ feature \xrightarrow{k_*} & IN\ class \\[4pt]
source \xrightarrow{f_s} & source\ feature \xrightarrow{k_s} & source\ class
\end{array}
$$

The ImageNet and source model are optimized for the accuracy of ImageNet estimates $\hat{p}(y_*|x_*)$ and source domain estimates $\hat{p}(y_s|x_s)$, respectively. Their accuracy on target domain estimates $\hat{p}(y_t|x_t)$ depends on:

1. Similarity between training and target inputs (i.e. images);
2. Robustness of input-to-feature mapping against training versus target input differences (covariate shift);
3. Similarity between training and target outputs (i.e. class labels).

Pre-trained models can be advantageous in terms of the first two criteria because (1) the larger and more diverse pre-training dataset is not source-biased and may better capture the target input distribution, (2) modern state-of-the-art network architectures are designed to learn more robust input-to-feature mappings, enabling better transfer to target tasks. However, since pre-training and target label space differ, the pre-trained classifier k_* needs to be replaced. One advantage of the source model is that it is trained for the target label space, but it may suffer from a lack of generalization to different input distributions.

Examples of target generalizability of pre-trained networks. Firstly, in Table 1, an example where target domain class-discriminative information is lost after source training is the Clipart-to-Real World ($C \rightarrow R$) transfer in Office-Home dataset. The target domain is more distant in style to the source domain than to ImageNet, such that oracle target accuracy (computed by fitting classifier head on the feature extractor using fully-labeled target samples) drops from 86.0% to 83.3% after source training. We refer readers to Section 5 and 6 for experiment details.

Secondly, the choice of pre-trained model can improve the robustness of input-to-feature mapping against covariate shift. In Table 1 for Office-Home dataset, ImageNet ResNet-101 has higher oracle target accuracy than ImageNet ResNet-50. Figure 4 visually compares VisDA-C target domain features extracted at the last pooling layer before the classification head of several models. Swin [31]

Pre-trained feat. extr.	C → R		A → C		R → A	
	Oracle	Co-learn	Oracle	Co-learn	Oracle	Co-learn
Source ResNet-50	83.3	74.1	**69.5**	53.9	*81.8*	70.5
ImageNet-1k ResNet-50	*86.0*	79.4	65.5	51.8	81.2	*71.1*
ImageNet-1k ResNet-101	**87.0**	**80.4**	68.0	**54.6**	**82.5**	**72.4**

Table 1: Oracle target accuracy of pre-trained feature extractor, and classification accuracy of adapted model co-learned with pre-trained feature extractor, on Office-Home.

(a) ImageNet-1k ResNet-101   (b) Source ResNet-101

(c) ImageNet-1k Swin-B   (d) Adapted ResNet-101

● plane ● bike ● bus ● car ● horse ● knife
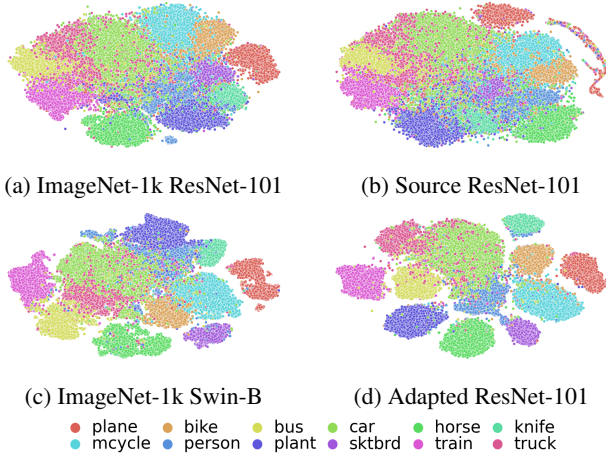● mcycle ● person ● plant ● sktbrd ● train ● truck

Figure 4: t-SNE visualization of VisDA-C target domain features by ImageNet-1k ResNet-101, source ResNet-101, ImageNet-1k Swin-B and source ResNet-101 adapted by proposed co-learning with ImageNet-1k Swin-B. Features are extracted at the last pooling layer before the classification head. Samples are colored by class.

is a recent transformer architecture with strong representation learning ability. Even without training on VisDA-C images, the ImageNet Swin-B feature extractor produces more class-discriminative target representations than both ImageNet and source ResNet-101 feature extractors.

4. Proposed Strategy

From our observations in Section 3, we propose to distil effective target domain information in pre-trained networks to generate improved pseudolabels to finetune the source model during target adaptation.

4.1. Preliminaries

We denote the source and target distributions as $\mathcal{P}_s$ and $\mathcal{P}_t$, and observations as $\mathcal{D}_s = \{(x_s^n, y_s^n)\}_{n=1}^{N_s}$ and $\mathcal{D}_t = \{(x_t^n, y_t^n)\}_{n=1}^{N_t}$, for image $x \in \mathcal{X}$ and one-hot classification label $y \in \mathcal{Y}$. The two domains share the same label space $\mathcal{Y}_s = \mathcal{Y}_t$ with $|\mathcal{Y}| = L$ classes, but have different data space $\mathcal{X}_s \neq \mathcal{X}_t$. In SFDA setting, source data $\mathcal{D}_s$ and target labels $\{y_t^n\}_{n=1}^{N_t}$ are not accessible during adaptation. Knowledge on source data is captured by a source model.

The source model trained on $\mathcal{D}_s$ is composed of feature

$\hat{\mathbf{y}}_{\mathbf{a}} = \hat{\mathbf{y}}_*$	$\mathbf{Conf}(\hat{\mathbf{y}}_{\mathbf{a}}) > \gamma$	$\mathbf{Conf}(\hat{\mathbf{y}}_*) > \gamma$	**Pseudolabel $\tilde{\mathbf{y}}$**
✓	✓ / ✗	✓ / ✗	$\hat{y}_a$
✗	✓	✓	-
✗	✓	✗	$\hat{y}_a$
✗	✗	✓	$\hat{y}_*$
✗	✗	✗	-

Table 2: MatchOrConf pseudolabeling scheme with adaptation and pre-trained model predictions, $\hat{y}_a$ and $\hat{y}_*$, and confidence threshold γ. Dash (-) means no pseudolabel.

extractor f_s parameterized by Θ_s that yields learned representations $z_s = f_s(x; \Theta_s)$, and classifier k_s parameterized by Ψ_s that yields logits $g_s = k_s(z_s; \Psi_s)$. Estimated class probabilities are obtained by $p_s = \sigma(g_s)$ for softmax function σ where $p_s[i] = \frac{\exp(g_s[i])}{\sum_{j=1}^{L} \exp(g_s[j])}$ for class i, and the predicted class is $\hat{y}_s = \arg\max_i p_s[i]$.

For a hypothesis $h \in \mathcal{H}$, we refer to $\epsilon_t(h, \ell_t) = E_{x \sim P_t} \epsilon(h(x), \ell_t(x))$ as the target risk of h with respect to the true target labeling function ℓ_t. To understand the relationship of h and l_t with the source model, we assume an error function ϵ such as $\epsilon(v, v') = |v - v'|^\alpha$ that satisfies triangle equality following [3, 4], then

$$\epsilon_t(h, \ell_t) \leq \epsilon_t(h, h_p) + \epsilon_t(h_p, \ell_t) \qquad (1)$$

where h_p is a pseudolabeling function. The second term $\epsilon_t(h_p, \ell_t)$ is target pseudolabeling error by h_p, and the first term is minimized by $h = h_p$. Motivated by the bound in Equation 1, we propose an iterative strategy to improve the pseudolabeling function h_p and to finetune h towards h_p. Our pseudolabeling function leverages a pre-trained network, which is discarded after target adaptation.

4.2. Co-learning with pre-trained networks

We propose a co-learning strategy to progressively adapt the source model $\{f_s, k_s\}$ with the pre-trained feature extractor f_*. The framework consists of two branches: (1) adaptation model branch $\{f_a, k_a\}$ initialized by source model $\{f_s, k_s\}$, (2) pre-trained model branch initialized by f_* and a newly-estimated task classifier q_*. Inspired by the nearest-centroid-classifier (NCC) in [28], we construct q_* as a weighted nearest-centroid-classifier where the centroid μ_i for class i is the sum of f_* features, weighted by estimated class probabilities of the adaptation model:

$$\mu_i = \frac{\sum_x \sigma(k_a(f_a(x; \Theta_a); \Psi_a))[i] f_*(x; \Theta_*)/\|f_*(x; \Theta_*)\|}{\sum_x \sigma(k_a(f_a(x; \Theta_a); \Psi_a))[i]} \qquad (2)$$

$$g_*[i] = q_*(f_*(x; \Theta_*))[i] = \frac{f_*(x; \Theta_*) \cdot \mu_i}{\|f_*(x; \Theta_*)\| \|\mu_i\|} \qquad (3)$$

$$p_* = \sigma(g_*/Temperature) \qquad (4)$$

The weighted NCC leverages the target domain class-discriminative cluster structures in f_* features. In Equa-

Algorithm 1 Proposed co-learning strategy

Input: Target images $\{x_t^n\}_{n=1}^{N_t}$; source model $k_s \circ f_s$ parameterized by (Θ_s, Ψ_s); pre-trained feature extractor f_* parameterized by Θ_*; confidence threshold γ; learning rate η; # episodes I;

1: **procedure** CO-LEARNING
2:      Initialize weights of adaption model branch $k_a \circ f_a$ by $(\Theta_a, \Psi_a) \leftarrow (\Theta_s, \Psi_s)$
3:      Initialize pre-trained model branch classifier q_* by computing centroids according to Equation 2
4:      **for** episode $i = 1 : I$ **do**
5:          Construct pseudolabel dataset $\tilde{\mathcal{D}}_t$ by MatchOrConf scheme with threshold γ in Table 2
6:          Compute loss $L_{co-learn}(\Theta_a)$ in Equation 5
7:          Update weights $\Theta_a \leftarrow \Theta_a - \eta \nabla L_{co-learn}(\Theta_a)$
8:          Update centroids in q_* according to Equation 2
9:      **end for**
10: **end procedure**
11: **return** Adaptation model $k_a \circ f_a$ with weights (Θ_a, Ψ_a)

---

tion 4, predictions p_* are sharpened with $Temperature = 0.01$ since the logits g_* calculated through cosine similarity are bounded in $[-1, 1]$

Co-learning: We alternate updates on the adaptation and pre-trained model branch as each adapts to the target domain and produces more accurate predictions to improve the other branch. Each round of update on the two branches is referred to as a co-learning episode. For the adaptation model branch, we freeze classifier k_a and update feature extractor f_a to correct its initial bias towards the source domain. We finetune f_a with cross-entropy loss on improved pseudolabeled samples, further described in the next paragraph. For the pre-trained model branch, we freeze feature extractor f_* to retain the target-compatible features and update classifier q_*. Since f_* is not updated throughout the adaptation process, only a single pass of target data through f_* is needed to construct a feature bank; previous works similarly required memory banks of source model features [50, 49, 48]. The centroids are updated by Equation 2 using estimated class probabilities $\sigma(k_a(f_a(x; \Theta_a); \Psi_a))$ of the current adaptation model. Improvements in the adaptation model predictions each episode give more accurate class centroids in the pre-trained model branch.

Improved pseudolabels: We improve pseudolabels each episode by combining predictions from the two branches using the MatchOrConf scheme in Table 2. Denoting $\hat{y}_a$ and $\hat{y}_*$ as the adaptation and pre-trained model predictions, respectively, the pseudolabel $\tilde{y}$ given is $\hat{y}_a (= \hat{y}_*)$ if the predicted classes match, and the predicted class with higher confidence level otherwise. Confidence level is determined by a threshold γ. Remaining samples are not pseudolabeled.

The co-learning objective for the adaptation model is

$$L_{co-learn}(\Theta_a) = - \sum_{(x, \tilde{y}) \in \tilde{\mathcal{D}}_t} \tilde{y} \cdot \log(p_a) \quad (5)$$

where $p_a = \sigma(k_a(f_a(x; \Theta_a); \Psi_a))$ and $\tilde{\mathcal{D}}_t$ is pseudolabeled target dataset. Algorithm 1 summarizes the strategy. We can incorporate the strategy into existing SFDA methods by replacing the pseudolabels used with our co-learned pseudolabels or by adding $L_{co-learn}$ to the learning objective.

5. Experiments and Results

We evaluate on 4 benchmark image classification datasets for domain adaptation. We describe experimental setups in Section 5.1 and results in Section 5.2.

5.1. Experimental setups

Datasets. Office-31 [41] has 31 categories of office objects in 3 domains: Amazon (A), Webcam (W) and DSLR (D). **Office-Home** [44] has 65 categories of everyday objects in 4 domains: Art (A), Clipart (C), Product (P) and Real World (R). **DomainNet** [36] is a challenging dataset with a total of 6 domains and 345 classes. Following [30], we evaluate on 4 domains: Clipart (C), Painting (P), Real (R) and Sketch (S) with a subset of 126 classes. **VisDA-C** [37] is a popular 12-class dataset for evaluating synthetic-to-real shift, with synthetic rendering of 3D models in source domain and Microsoft COCO real images in target domain. We report classification accuracy on the target domain for all domain pairs in Office-31, Office-Home and DomainNet, and average per-class accuracy on the real domain in VisDA-C. **Implementation details.** We follow the network architecture and training scheme in [28, 29] to train source models: Office-31, Office-Home and DomainNet use ResNet-50 and VisDA-C uses ResNet-101 initialized with ImageNet-1k weights for feature extractor plus a 2-layer linear classifier with weight normalization, trained on labeled source data. For our proposed strategy, we experiment with the following ImageNet-1k feature extractors curated on Torch [2] and Hugging Face [1] for co-learning: ResNet-50, ResNet-101, ConvNeXt-S, Swin-S, ConvNeXt-B and Swin-B, where S and B denote the small and base versions of the architectures, respectively. This list is not exhaustive and is meant as a demonstration that state-of-the-art networks can be successfully plugged into our proposed framework. ConvNeXt convolutional networks [32] and Swin transformers [31] are recently-released architectures for computer vision tasks that demonstrated improved robustness to domain shifts [18]. During target adaptation, we train using SGD optimizer for 15 episodes, with batch size 50 and learning rate 0.01 decayed to 0.001 after 10 episodes. We set confidence threshold $\gamma = 0.5$ for Office-31, Office-Home and DomainNet and $\gamma = 0.1$ for VisDA-C, with

Method	SF	A→D	A→W	D→A	D→W	W→A	W→D	Avg
MDD [52]	✗	93.5	94.5	74.6	98.4	72.2	**100.0**	88.9
GVB-GD [7]	✗	95.0	94.8	73.4	98.7	73.7	**100.0**	89.3
MCC [16]	✗	95.6	95.4	72.6	98.6	73.9	**100.0**	89.4
GSDA [14]	✗	94.8	**95.7**	73.5	99.1	74.9	**100.0**	89.7
CAN [17]	✗	95.0	94.5	**78.0**	99.1	77.0	99.8	90.6
SRDC [43]	✗	**95.8**	**95.7**	76.7	**99.2**	**77.1**	**100.0**	**90.8**
Source Only	✓	81.9	78.0	59.4	93.6	63.4	98.8	79.2
Co-learn (w/ ResNet-50)	✓	93.6	90.2	75.7	98.2	72.5	99.4	88.3
Co-learn (w/ Swin-B)	✓	**97.4**	**98.2**	84.5	99.1	82.2	100.0	93.6
SHOT† [28]	✓	95.0	90.4	75.2	**98.9**	72.8	99.8	88.7
w/ Co-learn (w/ ResNet-50)		94.2	90.2	75.7	98.2	74.4	**100.0**	88.8 (↑ 0.1)
w/ Co-learn (w/ Swin-B)		**95.8**	**95.6**	78.5	98.9	**76.7**	99.8	**90.9** (↑ 2.2)
SHOT++† [29]	✓	95.6	90.8	76.0	98.2	74.6	**100.0**	89.2
w/ Co-learn (w/ ResNet-50)		95.0	90.7	76.4	97.7	74.9	99.8	89.1 (↓ 0.1)
w/ Co-learn (w/ Swin-B)		**96.6**	**93.8**	79.8	98.9	78.0	100.0	**91.2** (↑ 2.0)
NRC† [48]	✓	92.0	91.6	74.5	97.9	74.8	**100.0**	88.5
w/ Co-learn (w/ ResNet-50)		95.4	89.9	76.5	98.0	76.0	99.8	89.3 (↑ 0.8)
w/ Co-learn (w/ Swin-B)		**96.2**	**94.3**	79.1	98.7	78.5	**100.0**	**91.1** (↑ 2.6)
AaD† [50]	✓	94.4	93.3	75.9	98.4	76.3	99.8	89.7
w/ Co-learn (w/ ResNet-50)		96.6	92.5	77.3	98.9	76.6	99.8	90.3 (↑ 0.6)
w/ Co-learn (w/ Swin-B)		**97.6**	**98.7**	82.1	99.3	80.1	100.0	**93.0** (↑ 3.3)

Table 3: Office-31: 31-class classification accuracy of adapted ResNet-50. For proposed strategy, ImageNet-1k feature extractor used is given in parenthesis. SF denotes source-free. † denotes reproduced results.

Method	SF	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
GSDA [14]	✗	**61.3**	76.1	79.4	65.4	73.3	74.3	65.0	53.2	80.0	72.2	60.6	83.1	70.3
GVB-GD [7]	✗	57.0	74.7	79.8	64.6	74.1	74.6	65.2	55.1	81.0	74.6	59.7	84.3	70.4
RSDA [12]	✗	53.2	**77.7**	81.3	66.4	74.0	76.5	67.9	53.0	**82.0**	75.8	57.8	85.4	70.9
TSA [26]	✗	57.6	75.8	80.7	64.3	76.3	75.1	66.7	55.7	81.2	75.7	61.9	83.8	71.2
SRDC [43]	✗	52.3	76.3	81.0	**69.5**	76.2	78.0	**68.7**	53.8	81.7	76.3	57.1	85.0	71.3
FixBi [34]	✗	58.1	77.3	80.4	67.7	**79.5**	78.1	65.8	**57.9**	81.7	**76.4**	**62.9**	**86.7**	72.7
Source Only	✓	43.5	67.1	74.2	51.5	62.2	63.3	51.4	40.7	73.2	64.6	45.8	77.6	59.6
Co-learn (w/ ResNet-50)	✓	51.8	78.9	81.3	66.7	78.8	79.4	66.3	50.0	80.6	71.1	53.7	81.3	70.0
Co-learn (w/ Swin-B)	✓	**69.6**	**89.5**	**91.2**	**82.7**	**88.4**	**91.3**	**82.6**	68.5	**91.5**	**82.8**	**71.3**	**92.1**	**83.5**
SHOT† [28]	✓	55.8	79.6	82.0	67.4	77.9	77.9	67.6	55.6	81.9	73.3	59.5	84.0	71.9
w/ Co-learn (w/ ResNet-50)		56.3	79.9	82.9	68.5	79.6	78.7	68.1	54.8	82.5	74.5	59.0	83.6	72.4 (↑ 0.5)
w/ Co-learn (w/ Swin-B)		61.7	82.9	85.3	72.7	80.5	82.0	71.6	60.4	84.5	76.0	64.3	86.7	75.7 (↑ 3.8)
SHOT++† [29]	✓	57.1	79.5	82.6	68.5	79.5	78.6	68.3	56.1	82.9	74.0	59.8	85.0	72.7
w/ Co-learn (w/ ResNet-50)		57.7	81.1	84.0	69.2	79.8	79.2	69.1	57.7	82.9	73.7	60.1	85.0	73.3 (↑ 0.6)
w/ Co-learn (w/ Swin-B)		63.7	83.0	85.7	72.6	81.5	83.8	72.0	59.9	85.3	76.3	65.3	86.6	76.3 (↑ 3.6)
NRC† [48]	✓	58.0	79.3	81.8	70.1	78.7	78.7	63.5	57.0	82.8	71.6	58.2	84.3	72.0
w/ Co-learn (w/ ResNet-50)		56.1	80.3	83.0	70.3	81.3	80.9	67.7	53.9	83.7	72.5	57.9	83.4	72.6 (↑ 0.6)
w/ Co-learn (w/ Swin-B)		67.8	86.4	89.1	80.7	87.5	89.3	77.8	**68.8**	89.7	81.6	68.7	89.9	81.4 (↑ 9.4)
AaD† [50]	✓	58.7	79.8	81.4	67.5	79.4	78.7	64.7	56.8	82.5	70.3	58.0	83.3	71.8
w/ Co-learn (w/ ResNet-50)		57.7	80.4	83.3	70.1	80.1	80.6	66.6	55.5	84.1	72.1	57.6	84.3	72.7 (↑ 0.9)
w/ Co-learn (w/ Swin-B)		65.1	86.0	87.0	76.8	86.3	86.5	74.4	66.1	87.7	77.9	66.1	88.4	79.0 (↑ 7.2)

Table 4: Office-Home: 65-class classification accuracy of adapted ResNet-50. For proposed strategy, ImageNet-1k feature extractor used is given in parenthesis. SF denotes source-free. † denotes reproduced results.

further analysis in Section 6. We also apply our strategy on existing methods. For SHOT [28] and SHOT++ [29] with pseudolabeling components, we replace the pseudolabels used with co-learned pseudolabels. For NRC [48] and AaD [50] originally without pseudolabeling components, we add $0.3L_{co-learn}$ to the training objective where the coefficient 0.3 follows that in SHOT [28] and SHOT++ [29].

5.2. Results

We report results for co-learning with the ImageNet-1k network (ResNet-50 or ResNet-101) used for source model initialization and Swin-B in Table 3, 4, 5 and 6. Best performance in each set of comparison is **bolded**, and best performance for each source-target transfer is underlined. Full results on other architectures are in Appendix .

Reusing pre-trained network from source model initialization: Reusing the same ImageNet network from source

Method	SF	C→P	C→R	C→S	P→C	P→R	P→S	R→C	R→P	R→S	S→C	S→P	S→R	Avg
Source Only	✓	47.2	61.7	50.7	47.2	71.7	44.3	58.2	63.2	49.2	59.5	52.4	60.5	55.5
Co-learn (w/ ResNet-50)	✓	58.7	75.7	51.9	54.9	76.6	46.1	61.4	63.7	49.1	65.0	62.7	76.7	61.9
Co-learn (w/ Swin-B)	✓	**69.1**	**85.0**	**61.3**	**68.7**	**87.3**	**60.6**	**70.3**	**71.5**	**59.5**	**70.1**	**72.9**	**85.2**	**71.8**
SHOT† [28]	✓	62.0	78.0	59.9	63.9	78.8	57.4	68.7	67.8	57.7	71.5	65.5	76.3	67.3
w/ Co-learn (w/ ResNet-50)		62.4	78.0	60.1	63.8	79.2	57.3	68.0	67.6	58.2	71.5	65.7	75.9	67.3 (=)
w/ Co-learn (w/ Swin-B)		**64.8**	**82.3**	**63.1**	**68.9**	**84.0**	**62.7**	**72.3**	**70.6**	**61.7**	**74.0**	**69.2**	**83.6**	**71.4** (↑ 4.1)
SHOT++† [29]	✓	63.0	80.4	62.5	66.9	80.0	60.0	71.2	68.7	61.2	72.8	66.6	78.1	69.3
w/ Co-learn (w/ ResNet-50)		63.0	80.0	61.9	65.9	80.2	60.6	70.6	68.8	61.2	73.3	67.2	78.2	69.2 (↓ 0.1)
w/ Co-learn (w/ Swin-B)		**65.9**	**82.7**	**65.3**	**71.0**	**84.0**	**64.9**	**73.5**	**71.7**	**64.6**	**75.7**	**70.6**	**84.5**	**72.9** (↑ 3.6)
AaD† [50]	✓	62.5	78.7	59.4	65.3	79.9	61.3	69.3	68.6	57.1	72.9	67.5	77.4	68.3
w/ Co-learn (w/ ResNet-50)		63.5	79.7	62.1	66.4	81.5	63.0	72.3	69.7	60.7	74.5	69.3	79.8	70.2 (↑ 1.9)
w/ Co-learn (w/ Swin-B)		**66.8**	**81.0**	**63.8**	**70.4**	**84.0**	**65.4**	**74.6**	**72.1**	**63.8**	**76.4**	**71.2**	**82.8**	**72.7** (↑ 4.4)

Table 5: DomainNet-126: 126-class classification accuracy of adapted ResNet-50. For proposed strategy, ImageNet-1k feature extractor used is given in parenthesis. SF denotes source-free. † denotes reproduced results.

Method	SF	plane	bike	bus	car	horse	knife	mcycle	person	plant	sktbrd	train	truck	Avg
SFAN [47]	✗	93.6	61.3	84.1	70.6	94.1	79.0	91.8	79.6	89.9	55.6	89.0	24.4	76.1
MCC [16]	✗	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
STAR [33]	✗	95.0	84.0	84.6	73.0	91.6	91.8	85.9	78.4	94.4	84.7	87.0	42.2	82.7
SE [10]	✗	95.9	87.4	85.2	58.6	96.2	95.7	90.6	80.0	94.8	90.8	88.4	47.9	84.3
CAN [17]	✗	**97.0**	87.2	82.5	74.3	97.8	**96.2**	90.8	80.7	96.6	**96.3**	87.5	**59.9**	87.2
FixBi [34]	✗	96.1	87.8	**90.5**	**90.3**	96.8	95.3	92.8	**88.7**	97.2	94.2	90.9	25.7	87.2
Source Only	✓	51.5	15.3	43.4	75.4	71.2	6.8	85.5	18.8	49.4	46.4	82.1	5.4	45.9
Co-learn (w/ ResNet-101)	✓	96.5	78.9	77.5	75.7	94.6	95.8	89.1	77.7	90.5	91.0	86.2	**51.5**	83.7
Co-learn (w/ Swin-B)	✓	**99.0**	**90.0**	84.2	81.0	**98.1**	97.9	**94.9**	80.1	94.8	95.9	94.4	48.1	88.2
SHOT† [28]	✓	95.3	87.1	79.1	55.1	93.2	95.5	79.5	79.6	91.6	89.5	87.9	56.0	82.4
w/ Co-learn (w/ ResNet-101)		94.9	84.8	77.7	63.0	94.1	95.6	85.6	81.0	**93.0**	92.2	86.4	60.4	84.1 (↑ 1.7)
w/ Co-learn (w/ Swin-B)		**96.0**	**88.1**	81.0	63.0	94.3	95.9	87.1	**81.8**	92.8	91.9	90.1	**60.5**	**85.2** (↑ 2.8)
SHOT++† [29]	✓	94.5	88.5	90.4	84.6	97.9	**98.6**	91.9	81.8	96.7	91.5	93.8	31.3	86.8
w/ Co-learn (w/ ResNet-101)		97.9	88.6	86.8	**86.7**	97.9	**98.6**	92.4	83.6	97.4	92.5	94.4	32.5	87.4 (↑ 0.6)
w/ Co-learn (w/ Swin-B)		**98.0**	**91.1**	88.6	83.2	97.8	97.8	92.0	**85.8**	**97.6**	93.2	**95.0**	43.5	**88.6** (↑ 1.8)
NRC† [48]	✓	96.8	**92.0**	83.8	57.2	96.6	95.3	84.2	79.6	**94.3**	93.9	90.0	59.8	85.3
w/ Co-learn (w/ ResNet-101)		96.9	89.2	81.1	65.5	96.3	96.1	**89.8**	80.6	93.7	**95.4**	88.8	60.0	86.1 (↑ 0.8)
w/ Co-learn (w/ Swin-B)		**97.4**	91.3	**84.5**	65.8	**96.9**	97.6	88.8	**82.0**	93.8	94.7	91.1	**61.6**	**87.1** (↑ 1.8)
AaD† [50]	✓	96.9	**90.2**	**85.7**	82.8	97.4	96.0	89.7	83.2	**96.8**	94.4	90.8	49.0	87.7
w/ Co-learn (w/ ResNet-101)		**97.7**	87.9	84.8	79.6	**97.6**	97.5	92.4	83.7	95.3	94.2	90.3	**57.4**	88.2 (↑ 0.5)
w/ Co-learn (w/ Swin-B)		97.6	**90.2**	85.0	**83.1**	**97.6**	97.1	92.1	**84.9**	**96.8**	95.1	92.2	56.8	**89.1** (↑ 1.4)

Table 6: VisDA-C: 12-class classification accuracy of adapted ResNet-101. For proposed strategy, ImageNet-1k feature extractor used is given in parenthesis. SF denotes source-free. † denotes reproduced results.

model initialization for co-learning can improve the original source model performance. Since Office domains have realistic images of objects similar to ImageNet, incorporating co-learning with SHOT and SHOT++ does not affect performance. NRC and AaD performance increase by 0.8% and 0.6%, respectively. Several Office-Home and VisDA-C domains contain images of a non-realistic style (e.g. Art, Clipart, Synthetic), and benefit more from co-learning with ImageNet networks. Existing methods improve by 0.5 − 0.9% on Office-Home, and 0.5 − 1.7% on VisDA-C. For the challenging DomainNet, SHOT and SHOT++ accuracy are little changed, and AaD accuracy increases by 1.9%.

Using more robust pre-trained network: Co-learning with the stronger and more robust ImageNet Swin-B significantly improves performance in all set-ups tested. Existing methods improve by 2.0 − 3.3% on Office, 3.6 − 9.4% on Office-Home, 3.6 − 4.4% on DomainNet, and 1.4 − 2.8% on VisDA-C. Interestingly, on Office and Office-Home, co-learning with ImageNet Swin-B alone is superior to integrating it with the existing SFDA methods tested, which have self-supervised learning components besides pseudolabeling. This implies that effective target domain information in source models is limited, and over-reliance on these models can impede target adaptation. In contrast, powerful and robust pre-trained networks can have features that are more class-discriminative on target domains, and the large performance boost reflects the advantage of using them to adapt the source models.

6. Further Analysis

We conduct further experiments with our proposed strategy in Section 6.1 and 6.2, and discuss considerations on

Pseudolabel strategy	A → C	A → P	A → R	Avg
SelfConf	49.1	74.8	77.1	67.0
OtherConf	57.5	*85.4*	*86.7*	76.5
Match	**61.3**	84.2	86.0	*77.2*
MatchOrConf	59.7	**86.3**	**87.1**	**77.7**
MatchAndConf	*60.3*	81.3	84.5	75.4

(a) Pseudolabeling strategies, on Office-Home

Confidence threshold γ	Office-31	Office-Home	VisDA
0.1	90.5	76.1	**87.1**
0.3	**91.2**	77.1	*86.8*
0.5	*91.0*	*77.4*	86.4
0.7	90.8	**77.5**	85.8
0.9	90.8	77.3	85.5
Target-compatibility ratio	0.982	0.947	0.844

(b) MatchOrConf confidence threshold. Target-compatibility ratio estimated by source to pre-trained model oracle target accuracy.

Table 7: Co-learning experiments with ImageNet-1k ConvNeXt-S in pre-trained model branch.

the use of pre-trained networks in Section 6.3.

6.1. Co-learning with pre-trained networks

We fix the pre-trained model branch to use ImageNet-1k ConvNeXt-S, which has an intermediate parameter size in our networks tested. Additional experiments on the pre-trained model branch and detailed results are in Appendix.
Adaptation model branch. We experiment with other pseudolabeling strategies besides MatchOrConf with a subset of domain pairs from Office-Home dataset in Table 7a: SelfConf selects confident samples from adaptation model branch, OtherConf selects confident samples from pre-trained model branch, Match selects samples with same predictions on both branches, and MatchAndConf selects confident samples with same predictions on both branches. SelfConf has the worst performance as source model confidence is not well-calibrated on target domain. Overall, MatchOrConf is the best strategy. From Table 7b, the optimal confidence threshold γ differs across datasets. We estimate the ratio of source to ImageNet feature extractor's target-compatibililty using the ratio of oracle target accuracy, computed by fitting a nearest-centroid-classifier head using fully-labeled target samples for each of the two feature extractors. When the ratio is low as in VisDA-C, the ImageNet feature extractor is more target-compatible, and lowering γ allows it to pseudolabel more samples according to its predictions. In practice, we set $\gamma = 0.1$ for VisDA-C as the Real target domain is more similar to ImageNet than to Synthetic source even by visual inspection, and default to $\gamma = 0.5$ otherwise although it may not be the optimal value.

Training curves. Figure 5 visualizes the co-learning process for VisDA-C. The proportion of target pseudolabels
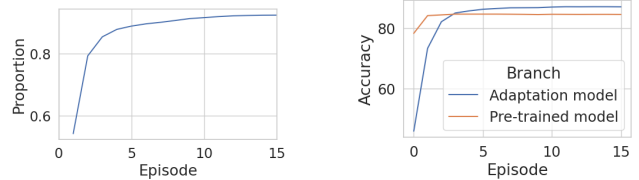
(a) Pseudolabel proportion   (b) Classification accuracy

Figure 5: VisDA-C co-learning training curves, with ImageNet-1k ConvNeXt-S in pre-trained model branch..

increases from 0.542 to 0.925 over 15 episodes. Classification accuracy on pre-trained model branch starts higher as the ImageNet feature extractor is more target-compatible than source feature extractor, and flattens earlier since its features are fixed. The adaptation model learns from pre-trained model predictions, and surpasses its accuracy as the feature extractor continues to adapt.

6.2. Other adaptation scenarios

Non-closed-set settings. Our co-learning strategy works well even under label shift where source and target label spaces do not match, as show in Table 8 on Office-Home. For open-set setting, the first 25 classes are target private, and the next 40 are shared. For partial-set setting, the first 25 classes are shared, and the next 40 are source private. For open-partial setting, the first 10 classes are shared, the next 5 are source private, and the remaining 50 are target private, and we report the harmonic mean of known and unknown class accuracy (H-score) following [51]. We add cross-entropy loss with co-learned pseudolabels at temperature 0.1 on open-set and partial-set version of SHOT and open-partial method OneRing [51].
Multi-source adaptation. Our proposed strategy can work with multi-source SFDA method as well. In Table 9 on Office-31, incorporating co-learned pseudolabels improves performance of CAiDA [9].

6.3. Discussion

We further explore considerations on the characteristics of the pre-trained networks used for co-learning and the effectiveness of our proposed strategy.

What are preferred characteristics of the feature extractor for co-learning? We first study this in Figure 6 through the relationship of ImageNet-1k top-1 accuracy and Office-Home average target domain accuracy. In general, a feature extractor with higher ImageNet accuracy has learned better representations and more robust input-to-feature mappings, and hence is likely to have higher oracle accuracy on target domain and consequently be more helpful in adapting the source model through co-learning. Next, we study a few specific source-domain pairs in Table 1 by plugging source and ImageNet-1k feature extractors into the pre-trained model branch, and observe effects of the following

Method	A → C	A → P	A → R	C → A	C → P	C → R	P → A	P → C	P → R	R → A	R → C	R → P	Avg
Open-set SHOT [28]	52.8	74.5	77.1	**57.6**	71.8	74.2	51.2	45.0	76.8	61.8	57.2	81.7	65.1
w/ Co-learn (w/ ResNet-50)	**55.0**	76.1	78.2	55.0	**73.7**	73.7	53.2	47.6	77.7	61.8	57.5	82.2	66.0 (↑ 0.9)
w/ Co-learn (w/ Swin-B)	52.2	**77.4**	**78.6**	56.3	71.3	**75.3**	**53.9**	**49.9**	77.9	62.5	58.2	82.3	66.3 (↑ 1.2)
Partial-set SHOT [28]	65.9	**86.1**	91.4	74.6	73.6	85.1	77.3	62.9	90.3	81.8	64.9	85.6	78.3
w/ Co-learn (w/ ResNet-50)	63.9	84.7	91.8	77.6	73.6	84.5	77.4	64.8	**90.5**	81.4	65.1	**87.8**	78.6 (↑ 0.3)
w/ Co-learn (w/ Swin-B)	**66.2**	85.9	**93.2**	78.2	74.7	85.6	81.5	65.8	90.4	82.6	65.4	87.7	**79.8** (↑ 1.5)
Open-partial OneRing [51]	**68.1**	81.9	87.6	72.2	76.9	83.3	80.7	**68.8**	88.2	80.5	66.0	85.5	78.3
w/ Co-learn (w/ ResNet-50)	64.9	**86.8**	88.0	71.9	76.7	83.5	**82.2**	67.6	**89.2**	82.8	**70.0**	86.4	**79.2** (↑ 0.9)
w/ Co-learn (w/ Swin-B)	65.8	85.9	88.0	72.8	77.9	83.9	81.9	65.8	88.5	82.9	67.2	**87.1**	79.0 (↑ 0.7)

Table 8: Office-Home: Accuracy for open-set and partial-set setting, H-score for open-partial setting, of adapted ResNet-50.

Method	→ A	→ D	→ W	Avg
CAiDA [9]	75.7	98.8	93.2	89.2
w/ Co-learn (w/ ResNet-50)	76.8	99.0	93.2	89.7 (↑ 0.5)
w/ Co-learn (w/ Swin-B)	**79.3**	**99.6**	**97.4**	**92.1** (↑ 2.9)

Table 9: Multi-source Office-31 accuracy of ResNet-50.
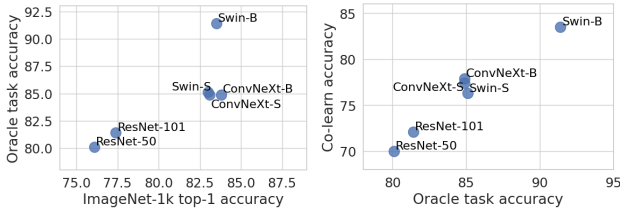
(a) Oracle target accuracy versus ImageNet-1k top-1 accuracy

(b) Accuracy after co-learning versus oracle target accuracy

Figure 6: Evaluations of ImageNet-1k networks. Oracle target accuracy of ImageNet-1k feature extractor, and accuracy of adapted model after co-learning with ImageNet-1k feature extractor, are evaluated on Office-Home.

Model	A → C	A → P	A → R	Avg
ConvNext-B + classification head†	56.1	78.6	83.3	72.7
ResNet-50 adapted w/ ConvNeXt-B	**60.5**	**86.2**	**87.3**	**78.0**
Swin-B + classification head†	67.1	86.7	89.1	81.0
ResNet-50 adapted w/ Swin-B	**69.6**	**89.6**	**91.2**	**83.5**

Table 10: Comparison of classification accuracy of ImageNet-1k feature extractor fitted with source-trained classification head, versus classification accuracy of adapted ResNet-50, on Office-Home. † denotes classifier is trained on fully-labeled source data.

positive characteristics: (1) Dataset similarity (input style and task): In $C \to R$ with the same ResNet-50 architecture, there are ImageNet samples more similar to Real World target than Clipart source is, and ImageNet-1k ResNet-50 has higher oracle accuracy than Source ResNet-50. (2) Robustness against covariate shift: In $A \to C$, although ImageNet is less similar to Clipart target than Art source is, changing from ResNet-50 to ResNet-101 improves oracle accuracy. (3) Different views: In $R \to A$, although ImageNet-1k ResNet-50 has slightly worse oracle accuracy than Source ResNet-50 (81.2% vs. 81.8%), co-learning with it has better adaptation performance (71.1% vs. 70.5%). Since the adaptation model branch is already initialized by source model, using ImageNet feature extractor in the other branch provides a different view of features and consequently classification decision which benefits co-learning. We refer readers to the Appendix for the full comparison on all 3 datasets.

Since modern ImageNet feature extractors have learned better representations, is it sufficient to use them in source models and not adapt? In Table 10, we fit a 2-layer linear classification head as in Section 5.1 on ImageNet-

1k ConvNext-B and Swin-B feature extractors by accessing and training the classifier on fully-labeled source data. On the Office-Home domain pairs tested, average performance of Swin-B + classification head is 8.3% higher than ConvNext-B + classification head, showing that feature extractor choice can indeed reduce negative effects of domain shift. However, adaptation to target domain is still necessary. Without having to access the source data and using a larger source model, Source ResNet-50 adapted with our proposed strategy achieves higher classification accuracy.

7. Conclusion

In this work, we explored the use of pre-trained networks beyond its current role as initializations in source training in SFDA. We observed that source-training can cause ImageNet networks to lose pre-existing generalization capability on target domain. We proposed distilling useful target domain information from pre-trained networks during target adaptation, and designed a simple co-learning strategy to improve target pseudolabel quality to finetune the source model. This framework also allows us to leverage modern pre-trained networks with improved robustness and generalizability. Experiments on benchmark datasets demonstrate the effectiveness of our framework and strategy.

Acknowledgments

References

[1] Hugging Face - models. https://huggingface.co/models. 5

[2] PyTorch - models and pre-trained weights. https://pytorch.org/vision/stable/models.html. 5

[3] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J.W. Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, 2010. 1, 4

[4] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman. Learning bounds for domain adaptation. In *NeurIPS*, 2007. 4

[5] W. Chen, L. Lin, S. Yang, D. Xie, S. Pu, Y. Zhuang, and W. Ren. Self-supervised noisy label learning for source-free unsupervised domain adaptation. *ArXiv*, 2021. 1, 3

[6] W. Chen, Z. Yu, S. D. Mello, S. Liu, J. M Alvarez, Z. Wang, and A. Anandkumar. Contrastive syn-to-real generalization. *ICLR*, 2021. 2

[7] S. Cui, S. Wang, J. Zhuo, C. Su, Q. Huang, and Q. Tian. Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*, 2020. 2, 6

[8] N. Ding, Y. Xu, Y. Tang, Y. Wang, and D. Tao. Source-free domain adaptation via distribution estimation. *CVPR*, 2022. 1, 3

[9] Jiahua Dong, Zhen Fang, Anjin Liu, Gan Sun, and Tongliang Liu. Confident anchor-induced multi-source free domain adaptation. In *NeurIPS*, 2021. 3, 8, 9

[10] G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for domain adaptation. In *ICLR*, 2018. 2, 7

[11] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17:59:1–59:35, 2016. 1, 2

[12] X. Gu, J. Sun, and Z. Xu. Spherical space domain adaptation with robust pseudo-label loss. In *CVPR*, 2020. 1, 2, 6

[13] I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. *ICLR*, 2021. 1

[14] L. Hu, M. Kan, S. Shan, and X. Chen. Unsupervised domain adaptation with hierarchical gradient synchronization. In *CVPR*, 2020. 1, 2, 6

[15] Ying Jin, Jiaqi Wang, and Dahua Lin. SepRep-Net: Multi-source free domain adaptation via model separation and reparameterization, 2023. 3

[16] Y. Jin, X. Wang, M. Long, and J. Wang. Minimum class confusion for versatile domain adaptation. In *ECCV*, 2020. 2, 6, 7

[17] G. Kang, L. Jiang, Y. Yang, and A. Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. *CVPR*, 2019. 1, 2, 6, 7

[18] D. Kim, K. Wang, S. Sclaroff, and K. Saenko. A broad study of pre-training for domain generalization and adaptation. *ArXiv*, 2022. 5

[19] Y. Kim, D. Cho, K. Han, P. Panda, and S. Hong. Domain adaptation without source data. *IEEE Transactions on Artificial Intelligence*, 2(6):508–518, 2021. 1, 3

[20] P. Koh, S. Sagawa, H. Marklund, S. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. Phillips, S. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang. Wilds: A benchmark of in-the-wild distribution shifts. *ArXiv*, 2020. 1

[21] J. N. Kundu, S. Bhambri, A. Kulkarni, H. Sarkar, V. Jampani, and R. V. Babu. Concurrent subsidiary supervision for unsupervised source-free domain adaptation. In *ECCV*, 2022. 3

[22] J. N. Kundu, A. Kulkarni, S. Bhambri, D. Mehta, S. Kulkarni, V. Jampani, and R. V. Babu. Balancing discriminability and transferability for source-free domain adaptation. In *ICML*, 2022. 3

[23] J. N. Kundu, N. Venkat, R. Ambareesh, R. M. V., and R. V. Babu. Towards inheritable models for open-set domain adaptation. *CVPR*, 2020. 1, 3

[24] H. Li, S. J. Pan, S. Wang, and A. C. Kot. Domain generalization with adversarial feature learning. *CVPR*, 2018. 1, 2

[25] R. Li, Q. Jiao, W. Cao, H.-S. Wong, and S. Wu. Model adaptation: Unsupervised domain adaptation without source data. *CVPR*, 2020. 1, 3

[26] S. Li, M. Xie, K. Gong, Chi H. Liu, Y. Wang, and W. Li. Transferable semantic augmentation for domain adaptation. In *CVPR*, 2021. 2, 6

[27] Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao. Deep domain generalization via conditional invariant adversarial networks. *ECCV*, 2018. 1, 2

[28] J. Liang, D. Hu, and J. Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. *ICML*, 2020. 1, 3, 4, 5, 6, 7, 9

[29] J. Liang, D. Hu, Y. Wang, R. He, and J. Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE TPAMI*, 2021. 1, 3, 5, 6, 7

[30] Mattia Litrico, Alessio Del Bue, and Pietro Morerio. Guiding pseudo-labels with uncertainty estimation for source-free unsupervised domain adaptation. In *CVPR*, 2023. 5

[31] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 3, 5

[32] Z. Liu, H. Mao, CY. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. *CVPR*, 2022. 5

[33] Z. Lu, Y. Yang, X. Zhu, C. Liu, YZ. Song, and T. Xiang. Stochastic classifiers for unsupervised domain adaptation. In *CVPR*, 2020. 2, 7

[34] J. Na, H. Jung, H. J. Chang, and W. Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *CVPR*, 2021. 2, 6, 7

[35] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019. 1

[36] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 5

[37] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *ArXiv*, abs/1710.06924, 2017. 5

[38] Z. Qiu, Y. Zhang, H. Lin, S. Niu, Y. Liu, Q. Du, and M. Tan. Source-free domain adaptation via avatar prototype generation and adaptation. *IJCAI*, 2021. 1, 3

[39] S. Qu, G. Chen, J. Zhang, Z. Li, W. He, and D. Tao. BMD: A general class-balanced multicentric dynamic prototype strategy for source-free domain adaptation. In *ECCV*, 2022. 3

[40] S. Roy, M. Trapp, A. Pilzer, J. Kannala, N. Sebe, E. Ricci, and A. Solin. Uncertainty-guided source-free domain adaptation. In *ECCV*, 2022. 3

[41] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. *ECCV*, 2010. 5

[42] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. *ECCV Workshops*, 2016. 1, 2

[43] H. Tang, K. Chen, and K. Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *CVPR*, 2020. 2, 6

[44] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. *CVPR*, 2017. 5

[45] G. Wilson and D. J. Cook. A survey of unsupervised deep domain adaptation. *ACM Trans. Intell. Syst. Technol.*, 11(5), July 2020. 1, 2

[46] H. Xia, H. Zhao, and Z. Ding. Adaptive adversarial network for source-free domain adaptation. *ICCV*, 2021. 1, 3

[47] R. Xu, G. Li, J. Yang, and L. Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *ICCV*, 2019. 1, 2, 7

[48] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. In *NeurIPS*, 2021. 1, 3, 5, 6, 7

[49] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui. Generalized source-free domain adaptation. In *ICCV*, 2021. 1, 3, 5

[50] S. Yang, Y. Wang, K. Wang, S. Jui, and J. van de Weijer. Attracting and dispersing: A simple approach for source-free domain adaptation. *NeurIPS*, 2022. 1, 3, 5, 6, 7

[51] Shiqi Yang, Yaxing Wang, Kai Wang, Shangling Jui, and Joost van de Weijer. One ring to bring them all: Towards open-set recognition under domain shift. *arXiv preprint arXiv:2206.03600*, 2022. 8, 9

[52] Y. Zhang, T. Liu, M. Long, and M. Jordan. Bridging theory and algorithm for domain adaptation. In *ICML*, 2019. 1, 2, 6